



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/982,950	10/22/2001	Junji Takagi	P14864-A	8293

466 7590 04/27/2004

YOUNG & THOMPSON
745 SOUTH 23RD STREET 2ND FLOOR
ARLINGTON, VA 22202

EXAMINER

COURTENAY III, ST JOHN

ART UNIT	PAPER NUMBER
----------	--------------

2126

DATE MAILED: 04/27/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/982,950

Applicant(s)

TAKAGI, JUNJI

Examiner

St. John Courtenay III

Art Unit

2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on 22 October 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 22 October 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All b) ☐ Some * c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 2.

- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.


ST. JOHN COURTENAY III
PRIMARY EXAMINER

Detailed Action

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1- 24 are rejected under 35 U.S.C. § 102(e) as being anticipated by **Waldo et al.** (U.S. Patent 6,708,171).

As per independent claim 1:

Waldo teaches a stub search loading system for, in executing remote method invocation from a plurality of clients to a server, downloading a stub necessary in a request source client from the server, wherein:

- the request source client comprises stub search means for sending a stub request formed from a stub name and client identifier to the server and receiving a stub returned from the server [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see also attempt to obtain the stub class instance for the server and associated discussion col. 9, beginning line 29], and

- the server comprises a stub search interface for, in response to the stub request from the request source client, returning to the request source client the stub appropriate for a runtime environment of the request source client on the basis of the designated stub name and client identifier [e.g., see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44; see also initiation of communications with the nameserver computer 13 to locate and load a stub class instance for the class to process the remote method and associated discussion col. 10 (nameserver discussion), col. 11, lines 16-40, lines 54-67].

As per independent claim 5:

This claim is rejected for the same reasons detailed above in the rejection of independent claim 1, and also for the following additional reasons:

Waldo teaches a stub search loading method of, in executing remote method invocation from a plurality of clients to a server, downloading a stub necessary in a request source client from the server, comprising the steps of:

- sending a stub request formed from a stub name and client identifier from the request source client to the server [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see also col. 9, lines 29-34];
- upon receiving the stub request, returning from the server to the request source client a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client, on the basis of the

designated stub name and client identifier [e.g., the "nameserver" provides identification, col. 9, line 54; see also initiation of communications with the nameserver computer 13 to locate and load a stub class instance for the class to process the remote method and associated discussion col. 10 (nameserver discussion), col. 11, lines 16-40, lines 54-67]; and

- receiving, at the request source client, the stub transmitted from the server [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per independent claim 9:

This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Waldo teaches a server apparatus for providing a stub necessary in executing remote method invocation to a request source client in response to a stub request from the client, comprising:

- a stub search interface for, in response to the stub request from the request source client, returning to the request source client the stub appropriate for a runtime environment of the request source client on the basis of the designated stub name and client identifier [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39].

As per independent claim 13:

This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Waldo teaches a client apparatus for downloading a stub necessary in a client in executing remote method invocation from a server, comprising:

- stub search means for transmitting a stub request formed from a stub name and client identifier to a server having a stub search interface for, in response a to the stub request from the client, returning to the request source client the stub appropriate for a runtime environment of the request source client on the basis of the designated stub name and client identifier [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39], and
- receiving the stub returned from the server apparatus in response to the stub request [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per independent claim 17:

This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Waldo teaches a computer-readable recording medium which stores a program for executing stub search loading processing of, in executing remote method invocation from a plurality of clients to a server, downloading a stub necessary in a request source client from the server, wherein the program comprises:

- a procedure code for sending a stub request formed from a stub name and client identifier from the request source client to the server [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID];
- a procedure code for, upon receiving the stub request, returning from the server to the request source client a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client, on the basis of the designated stub name and client identifier [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14; see also "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID]; and
- a procedure code for receiving, at the request source client, the stub transmitted from the server [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per independent claim 21:

This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Waldo teaches a computer-readable recording medium which stores a program for executing processing of providing a stub necessary in executing remote method invocation to a request source client in response to a stub request from the client, wherein the program comprises:

- a procedure code for, in response to the stub request from the request source client, returning to the request source client the stub appropriate for a runtime environment of the request source client on the basis of the designated stub name and client identifier [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44; see also Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID].

As per dependent claim 2:

Waldo teaches:

- the server further comprises a stub set in which a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client is prepared for each of types of the clients having different runtime environments [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and
- upon receiving the stub request from the request source client, the stub search interface searches the stub set for the corresponding stub on the basis of the designated stub name and client identifier and returns the stub to the request source client [see Unique service ID associated with

service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14; see also "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID].

As per dependent claim 3:

Waldo teaches:

- the server further comprises stub generation means for generating, for each of types of the clients having different runtime environments, a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and
- upon receiving the stub request from the client, the stub search interface returns to the request source client the stub appropriate for the runtime environment of the request source client, which is generated by the stub generation means on the basis of the designated stub name and client identifier [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14; see also "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID].

As per dependent claim 4:

Waldo teaches:

- the server further comprises stub generation means for generating, for each of types of the clients having different runtime environments, a stub to be used together with a skeleton used in the server at the time of remote method invocation from the client [e.g., see execution environment control module 19 and associated discussion col. 11, lines 9-67; see also "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and
- when the corresponding stub is not present in the stub set, the stub search interface returns to the request source client the stub appropriate for the runtime environment of the request source client, which is generated by the stub generation means on the basis of the designated stub name and client identifier [see "class not found" exception col. 11, lines 48-55, and alternative method of finding a resource located from the nameserver computer 13, col. 11, lines 55-67, discussion cont'd col. 12].

As per dependent claim 6:

Waldo teaches:

- the return step comprises the steps of in the server on the basis of the designated client identifier, selecting one of stub sets in which a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client is prepared for each of types of the clients having different runtime environments [e.g., see execution environment control module 19 and

associated discussion col. 11, lines 9-67; see also "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID],

- selecting the stub having the designated stub name from the selected stub set [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and,
- transmitting the selected stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 7:
Waldo teaches:

- the return step comprises the steps of in the server on the basis of the designated stub name and client identifier, generating a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and

- transmitting the generated stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 8:

Waldo teaches the return step comprises:

- in the server, when a stub cannot be selected from the stub set, generating a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client on the basis of the designated stub name and client identifier [see "class not found" exception col. 11, lines 48-55, and alternative method of finding a resource located from the nameserver computer 13, col. 11, lines 55-67, discussion cont'd col. 12], and
- transmitting the generated stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 10:

Waldo teaches:

- the apparatus further comprises a stub set in which a stub to be used together with a skeleton used at the time of remote method invocation from the request source client is prepared for each of types of clients having different runtime environments, including the request source client [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and

- upon receiving from the request source client the stub request formed from the stub name and client identifier, the stub search interface searches the stub set for the appropriate stub on the basis of the designated stub name and client identifier and returns the stub to the request source client [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14; see also "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID].

As per dependent claim 11:

Waldo teaches:

- the apparatus further comprises stub generation means for generating, for each of types of clients having different runtime environments, including the request source client, a stub to be used together with a skeleton used at the time of remote method invocation from the request source client [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and
- upon receiving the stub request from the client, the stub search interface returns to the request source client the stub appropriate for the runtime environment of the request source client, which is generated by the stub generation means on the basis of the designated stub name and client identifier [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14; see also "the server 12(m) returns the stub to the stub class loader on the client

computer (step 612)" and associated discussion, col. 17, lines 41-44, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID].

As per dependent claim 12:

Waldo teaches:

- the apparatus further comprises stub generation means for generating, for each of types of clients having different runtime environments, including the request source client, a stub to be used together with a skeleton used at the time of remote method invocation from the client [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and
- when the corresponding stub is not present in the stub set, the stub search interface returns to the request source client the stub appropriate for the runtime environment of the request source client, which is generated by the stub generation means on the basis of the designated stub name and client identifier [see "class not found" exception col. 11, lines 48-55, and alternative method of finding a resource located from the nameserver computer 13, col. 11, lines 55-67, discussion cont'd col. 12].

As per dependent claim 14:

Waldo teaches:

- the apparatus further comprises a stub set in which a stub to be used together with a skeleton used in the server apparatus at the time of remote method invocation from the request source client is prepared for each of types of clients having different runtime environments, including the request

source client [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and

- upon receiving from the request source client the stub request formed from the stub name and client identifier, the stub search interface searches the stub set for the appropriate stub on the basis of the designated stub name and client identifier and returns the stub to the request source client [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14; see also "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID].

As per dependent claim 15:

Waldo teaches:

- the apparatus further comprises stub generation means for generating, for each of types of clients having different runtime environments, including the request source client, a stub to be used together with a skeleton used in the server apparatus at the time of remote method invocation from the request source client [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and
- upon receiving the stub request from the client, the stub search interface returns to the request source client the stub appropriate for the runtime environment of the request source client, which is generated by the stub generation means on the basis of the designated stub name and client

identifier [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14; see also "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID].

As per dependent claim 16:

Waldo teaches:

- the apparatus further comprises stub generation means for generating, for each of types of clients having different runtime environments a stub to be used together with a skeleton used in the server apparatus at the time of remote method invocation from the client [see execution environment control module 19 and associated discussion col. 11, lines 9-67], and
- when the corresponding stub is not present in the stub set, the stub search interface returns to the request source client the stub appropriate for the runtime environment of the request source client, which is generated by the stub generation means on the basis of the designated stub name and client identifier [see "class not found" exception col. 11, lines 48-55, and alternative method of finding a resource located from the nameserver computer 13, col. 11, lines 55-67, discussion cont'd col. 12].

As per dependent claim 18:

Waldo teaches the program for executing the procedure code for returning comprises:

- a procedure code for, in the server on the basis of the designated client identifier, selecting one of stub sets in which a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client is prepared for each of types of the clients having different runtime environments [e.g., see execution environment control module 19 and associated discussion col. 11, lines 9-67; see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID],
- a procedure code for selecting the stub having the designated stub name from the selected stub set [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and
- a procedure code for transmitting the selected stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 19:

Waldo teaches the program for executing the procedure code for returning comprises:

- a procedure code for, in the server on the basis of the designated stub name and client identifier, generating a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and
- a procedure code for transmitting the generated stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 20:

Waldo teaches the program for executing the procedure code for returning comprises:

- a procedure code for, in the server, when a stub cannot be selected from the stub set, generating a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client on the basis of the designated stub name and client identifier [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also

col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and

- a procedure code for transmitting the generated stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 22:

Waldo teaches the program for executing the procedure code for returning comprises:

- a procedure code for, in the server on the basis of the designated client identifier, selecting one of stub sets in which a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client is prepared for each of types of the clients having different runtime environments [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID],
- a procedure code for selecting the stub having the designated stub name from the selected stub set [see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and

- a procedure code for transmitting the selected stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 23

Waldo teaches the program for executing the procedure code for returning comprises:

- a procedure code for, in the server on the basis of the designated stub name and client identifier, generating a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and
- a procedure code for transmitting the generated stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

As per dependent claim 24

Waldo teaches the program for executing the procedure code for returning comprises:

- a procedure code for, in the server, when a stub cannot be selected from the stub set, generating a stub to be used together with a skeleton used in the server at the time of remote method invocation from the request source client on the basis of the designated stub name and client identifier, [e.g., see "upon receipt of the request for the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service," col. 17, lines 35-39; see Unique service ID associated with service 400 and stub 404 and attributes 406, shown in fig 4, and discussed col. 17, lines 1-14, also col. 17, lines 27-34 regarding client's request in the form of a specific service ID], and,
- a procedure code for transmitting the generated stub from the server to the request source client [see "the server 12(m) returns the stub to the stub class loader on the client computer (step 612)" and associated discussion, col. 17, lines 41-44].

Application/Control Number:
09/982,950
Art Unit: 2126

Page 21

Prior Art not relied upon:

Please refer to the references listed on the attached PTO-892
which are not relied upon in the claim rejections detailed above.

Application/Control Number:
09/982,950
Art Unit: 2126

Page 22

How to Contact the Examiner:

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to **St. John Courtenay III** whose voice telephone number is **(703) 308-5217**. A voice mail service is also available at this number. Normal Flex work schedule: M – F 7:30 AM - 4:00 PM

- **All responses sent by U.S. Mail should be mailed to:**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Patent Customers advised to FAX communications to the USPTO

<http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/faxnotice.pdf>

Effective Oct. 15, 2003, ALL patent application correspondence transmitted by FAX must be directed to the new PTO central FAX number:


**NEW PTO CENTRAL FAX NUMBER:
703-872-9306**

-
- Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: (703) 305-3900.**

Please direct inquiries regarding fees, paper matching, and other issues not involving the Examiner to:

Technical Center 2100 CUSTOMER SERVICE: 703 306-5631

The Manual of Patent Examining Procedure (MPEP) is available online at:
<http://www.uspto.gov/web/offices/pac/mpep/index.html>


**ST. JOHN COURTENAY III
PRIMARY EXAMINER**